

# 光達雷達感測

潘柏宇, 陳政紹, 龔彥安

2023/1/1

boyupan289@gmail.com, chenzhengshao8@gmail.com, 110525010@cc.ncu.edu.tw

## 1 前言

本次實驗主的目的利用MCU(ESP32)來操作馬達(SG90)、距離感測器(VL53L0X)、蜂鳴器、Wifi(Blynk lib)實作光達雷達感測。MCU作為控制中樞藉以控制九軸感測器及距離感測器的操作，首先對周遭環境進行感測已收集初始環境距離。之後若是感測範圍內存在物體則觸發蜂鳴器，記錄物體距離、觸發時間並通報至手機。

## 2 系統設計

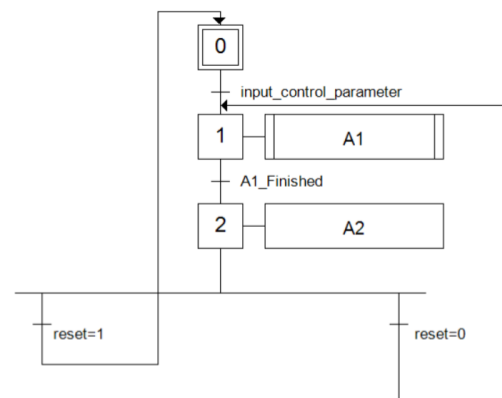
本專題利用 MIAT 方法進行系統設計，利用距離感測器能夠及時偵測距離的特性和馬達控制感測方向，去判斷範圍內是否有物體在移動。設定參數後系統開始運作，A11會控制馬達轉動方向，目前設置五個方向依序循環(後續也能在擴充更多種角度)。轉到特定角度之後會進到距離感測功能(A12)，此功能會去偵測並記錄此方向目前的距離。進入到A2後會進行距離的運算並將資料傳到後端系統當中。在開機的第一輪只會記錄每個方向的初始距離，在進入第二輪之後才會開始運算，會將上一輪所記錄的距離和目前偵測到的距離做比較，如果差距大於閾值則代表此方向有偵測到物體運動。如果偵測到有物體運動就會去改變後端的資料，將最新的數據更新顯示到手機當中。

## 3 硬體架構

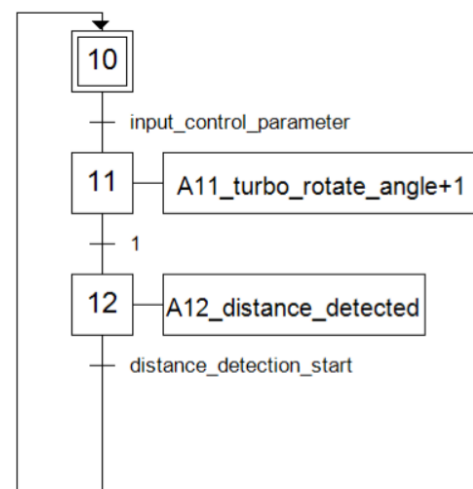
本節展示了本專題所用到的硬體說明與介面協定。

### 3.1 MCU

ESP32和ESP8266是系列低成本，低功耗的單晶片微控制器，整合了Wi-Fi和雙模藍牙。ESP32系列採Tensilica Xtensa LX6微處理器，包括雙核和單核變體，內建天線開關，RF變換器，功率放大器，低雜訊接收放



(a) 離散事件建模A0



(b) 離散事件建模A1

Figure 1: 離散事件建模

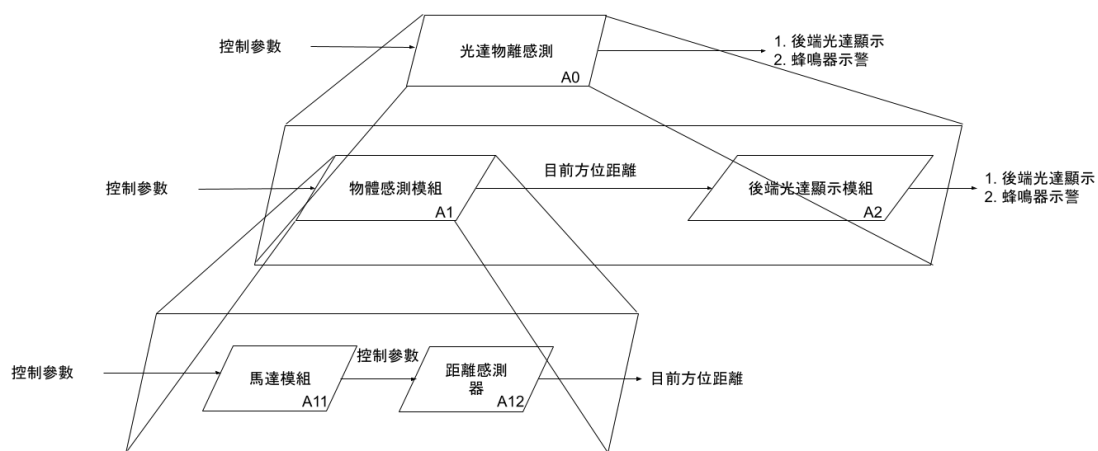


Figure 2: 系統設計

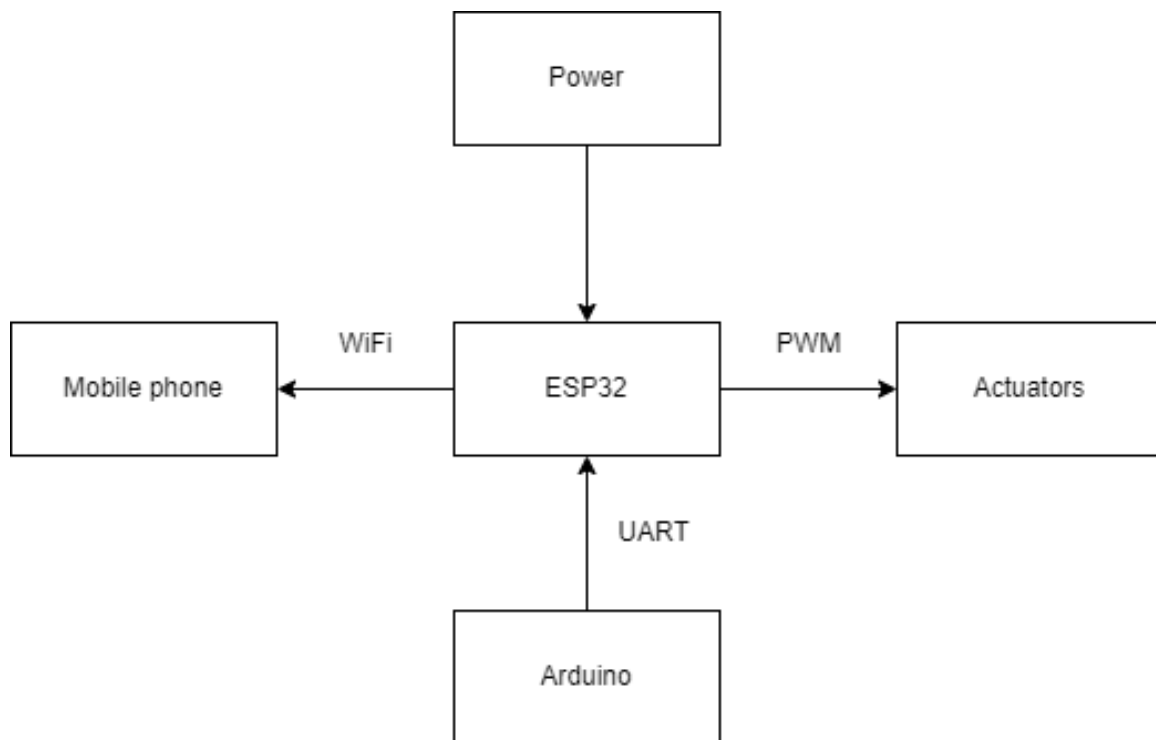


Figure 3: 硬體架構

器，濾波器和電源管理模組。本次專題採的是ESP32模組。

### 3.2 Actuators

- **Motor(SG90)**：伺服馬達裡含有直流馬達、齒輪箱、軸柄、以及控制電路，可透過訊號控制軸柄的停止角度，大概都是0到180度。一般伺服馬達有三條線，電源、接地、訊號線。透過訊號線傳送PWM脈波來控制軸柄的停止位置旋轉角度。
- **Buzzer**：有源蜂鳴器、無源蜂鳴器。無源蜂鳴器可透過 PWM 方式控制其發聲頻率，我們就可以利用它來播放簡單的旋律。有源蜂鳴器則是內建了一組固定的頻率，只要接通電源，就會發出固定的音調，無法利用 PWM 對其音頻進行控制。
- **Distance sensor(VL5310x)**：可直接判斷與目標物體的距離，最遠可達 2 公尺，回報的距離不受目標的反射率影響，在高度的紅外線環境光亮度下運作，先進嵌入式光學串音補償，能夠簡化玻璃外蓋的選擇，透過 vl5310x 感測器可觀測回傳前方物件距離。

### 3.3 介面協定

- **UART**: 使用 UART 燒錄程式碼到 MCU ESP32 當中。
- **PWM**: 使用 PWM 與周邊硬體溝通。

## 4 系統程式高階合成

我們使用 Arduino IDE 做為開發環境，並使用 C 語言開發 MCU 與其他 Actuators 溝通之邏輯。本節會介紹本專題的程式撰寫內容。

### 4.1 系統程式高階合成G0

```
void main()
{
    printf("X0=%d,X1=%d,X2=%d\n",X0,X1,X2);
    while(1)
    {
        datapath0();
        grafcet0();
        printf("X0=%d,X1=%d,X2=%d\n",X0,X1,X2);
    }
}
void grafcet0()
```

```
if((X0 == 1) && (
    input_control_parameter
))
{
    X0 = 0;
    X1 = 1;
    return;
}

if((X1 == 1) && (
    A1_Finished))
{
    X1 = 0;
    X2 = 1;
    return;
}

if(X2 == 1)
{
    if(reset=1)
    {
        X2 = 0;
        X0 = 1;
    }
    else if(reset=0)
    {
        X2 = 0;
        X1 = 1;
    }
    return;
}
```

```
}
void datapath0()
{
    if(X0 == 1)
        action();
    if(X1 == 1)
        A1();
    if(X2 == 1)
        A2();
}
```

### 4.2 系統程式高階合成G1

```
void A1()
{
    printf("A1 activate!!\n");
    datapath1();
    grafcet1();
}
```

```

printf("X10_=%d,X11_=%d,X12_=%d\n",X10 ,X11 ,X12 );
}
void datapath1 ()
{
    if(X10 == 1)
    A10_action();
    if(X11 == 1)
    A11_turbo_rotate_angle();
    if(X12 == 1)
    A12_distance_detected();

}
void grafcet1 ()
{

    if((X10 == 1) && (
        input_control_parameter
        ))
    {
        X10 = 0;
        X11 = 1;
        return;
    }

    if((X11 == 1) && (1))
    {
        X11 = 0;
        X12 = 1;
        return;
    }

    if((X12 == 1) && (
        distance_detection_start
        ))
    {
        X12 = 0;
        X10 = 1;
        return;
    }

}

```

重要程式碼參考: <https://github.com/Peter-Kung/LiDAR-Sensing>

## 5 軟體模擬

Figure 4 是我們的軟體模擬合成過程。

## 6 展示

Figure 5 、 6 與 7 是我們的成品展示。

## 7 主要人力研究

Table 1 是我們的主要人力研究配置細節。

```

16:20:44.766 -> X0 = 1,X1 = 0,X2 = 0
16:20:44.766 -> action activate !!
16:20:44.766 -> X0 = 0,X1 = 1,X2 = 0
16:20:44.766 -> A_one activate !!
16:20:44.766 -> X10 = 0,X11 = 1,X12 = 0
16:20:44.766 -> X0 = 0,X1 = 1,X2 = 0
16:20:44.766 -> A_one activate !!
16:20:45.110 -> A_one1_turbo_rotate_angle+1 activate !!
16:20:45.110 -> X10 = 0,X11 = 0,X12 = 1
16:20:45.110 -> X0 = 0,X1 = 1,X2 = 0
16:20:45.110 -> A_one activate !!
16:20:45.110 -> A_one2_distance_detected activate !!
16:20:45.676 -> X10 = 1,X11 = 0,X12 = 0
16:20:45.676 -> X0 = 0,X1 = 0,X2 = 1
16:20:45.676 -> A_two activate !!

```

Figure 4: 軟體模擬

學生	程式碼撰寫	軟硬體建置	MIAT 建構
潘柏宇	vl5310x 程式碼	環境設定	IDEF0 撰寫
陳政紹	SG90 程式碼	接腳位	Grafcet0 撰寫
龔彥安	Wifi 程式碼	接腳位	Grafcet1 撰寫

Table 1: 分配工作內容說明

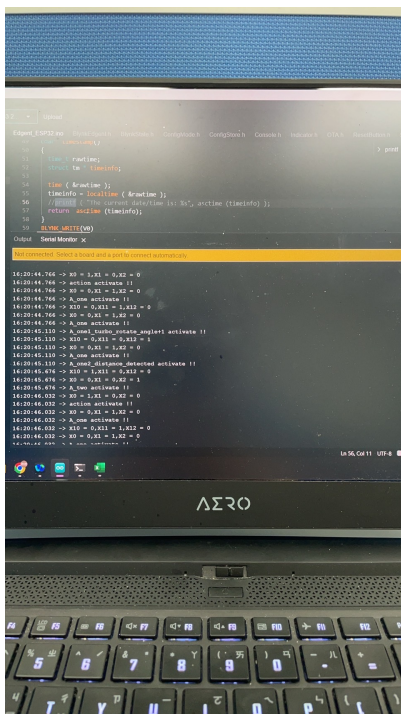


Figure 5: 展示1

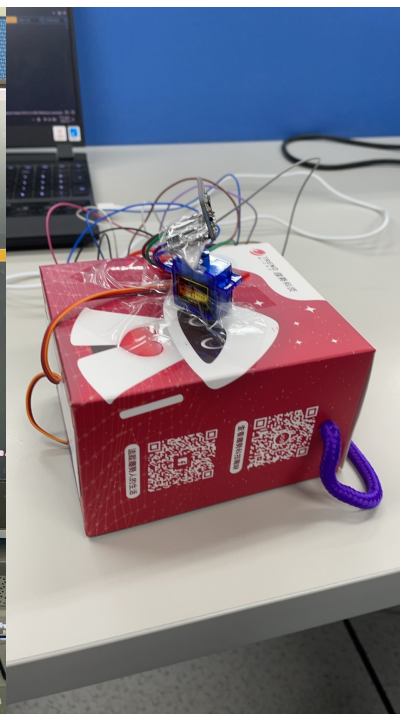


Figure 6: 展示2

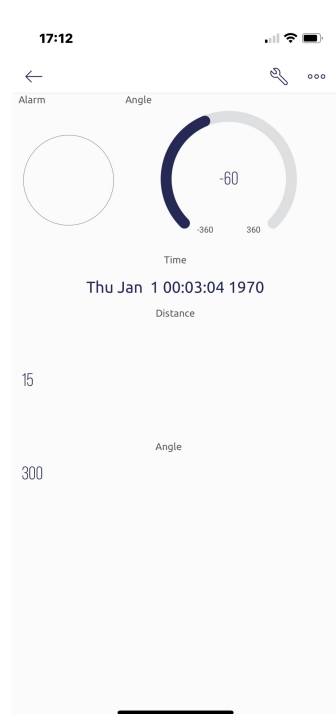


Figure 7: 展示3